



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06F 17/30336 (2006.01); G06F 17/30946 (2006.01)

(21)(22) Заявка: 2017109115, 20.03.2017

(24) Дата начала отсчета срока действия патента:
20.03.2017

Дата регистрации:
06.04.2018

Приоритет(ы):

(22) Дата подачи заявки: 20.03.2017

(45) Опубликовано: 06.04.2018 Бюл. № 10

Адрес для переписки:

109456, Москва, Рязанский пр-кт, 75, корп. 4, 1-
ая башня, 7 этаж, ООО "ФПБ "ГАРДИУМ"

(72) Автор(ы):

Семенов Алексей Петрович (RU)

(73) Патентообладатель(и):

Семенов Алексей Петрович (RU)

(56) Список документов, цитированных в отчете
о поиске: US 5442784 A, 15.08.1995. US
6505205 B1, 07.01.2003. US 6163775 A,
19.12.2000. RU 2364921 C2, 20.08.2009. RU
2613026 C1, 14.03.2017.

(54) ЭЛЕКТРОННАЯ БАЗА ДАННЫХ И СПОСОБ ЕЁ ФОРМИРОВАНИЯ

(57) Реферат:

Изобретение относится к области обработки данных. Техническим результатом является повышение надежности базы данных за счет исключения ошибок при добавлении новых данных и формировании запросов и отчетов, а также снижения риска лавинообразной деградации производительности из-за отсутствия необходимых индексов при работе с большими объемами данных. В способе формирования электронной базы данных данные организуют в единственную таблицу. Формируют строки, отображающие корневой элемент, тип данных, термин, реквизит термина и данные. Сохраняют первый идентификационный номер, уникальный идентификационный номер типа данных,

отличающийся от первого идентификационного номера, наименование используемого типа данных, код типа данных, уникальный идентификационный номер термина, наименование термина, код используемого типа данных, уникальный идентификационный номер реквизита термина, идентификационный номер родительского элемента реквизита термина, зависящего от родительского элемента, уникальный идентификационный номер данных, идентификационный номер родительского элемента, данные, зависящие от родительского элемента, и уникальный код типа данных. 1 з.п. ф-лы, 8 ил., 3 табл.

RU 2 650 032 C1

RU 2 650 032 C1



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(52) CPC

G06F 17/30336 (2006.01); *G06F 17/30946* (2006.01)(21)(22) Application: **2017109115, 20.03.2017**(24) Effective date for property rights:
20.03.2017Registration date:
06.04.2018

Priority:

(22) Date of filing: **20.03.2017**(45) Date of publication: **06.04.2018** Bull. № 10

Mail address:

**109456, Moskva, Ryazanskij pr-kt, 75, korp. 4, 1-aya
bashnya, 7 etazh, OOO "FPB "GARDIUM"**

(72) Inventor(s):

Semenov Aleksej Petrovich (RU)

(73) Proprietor(s):

Semenov Aleksej Petrovich (RU)(54) **ELECTRONIC DATABASE AND METHOD OF ITS FORMATION**

(57) Abstract:

FIELD: data processing.

SUBSTANCE: invention relates to the field of data processing. In the method of forming an electronic database, the data is organized into a single table. Form lines displaying the root element, data type, term, term props and data. First identification number, a unique identification number of the data type, different from the first identification number, the name of the data type used, the data type code, the unique identification number of the term, the term name, the code of the data type used, the unique identification number of the terminology, the identification number of the parent

element of the terminus that depends on the parent element, a unique data identification number, the identification number of the parent element, data that depends on the parent element, and a unique data type code.

EFFECT: technical result is to improve the reliability of the database by eliminating errors when adding new data and generating queries and reports, and also reduce the risk of avalanche-like performance degradation due to the lack of necessary indices when working with large amounts of data.

1 cl, 8 dwg, 3 tbl

RU 2 650 032 C1

RU 2 650 032 C1

ОБЛАСТЬ ТЕХНИКИ

Группа изобретений относится к области обработки данных, а именно к электронной базе данных и способу формирования электронной базы данных.

УРОВЕНЬ ТЕХНИКИ

- 5 Патент US 5369761 описывает систему, которая может быть использована для того, чтобы администратор базы данных мог выборочно денормализовать базу данных для пользователей и программистов. Система ведет запись преобразований между денормализованными полями и базовыми полями, из которых они были получены. Устройства обработки данных обращаются к этим записям, чтобы поддерживать
- 10 самосогласованность базы данных и по возможности извлекать данные из денормализованных полей.

СУЩНОСТЬ

- Реализация группы изобретений была разработана на основе оценки разработчиками по меньшей мере одной технической проблемы, связанной с техническими решениями
- 15 предшествующего уровня техники.

Группа изобретений является результатом исследований разработчика, что технические решения из уровня техники не всегда обеспечивают гибкость и нестабильно функционируют из-за непосредственного участия пользователя в построении структуры базы данных.

- 20 Группа изобретений является результатом исследований разработчика, что техническим решениям из уровня техники может потребоваться ручное задание соответствия полей нормализованных таблиц и денормализованной таблицы, предоставляя удобство ручной выборки данных ценой накладных расходов на хранение дублирующейся информации. Решение из уровня техники является решением одного
- 25 из фрагментов информационной системы, не формирующее законченную систему. Кроме того, это решение не позволяет работать с большими объемами данных без дополнительных действий над архитектурой приложения: дальнейшая нормализация, создание индексов и разделов данных (сегментов или партиций).

- Группа изобретений является результатом исследований разработчика технической
- 30 проблемы, решаемой группой изобретений и заключающейся в необходимости создания универсальной базы данных.

- Техническим результатом заявляемой группы изобретений является повышение надежности базы данных за счет исключения ошибок при добавлении новых данных и формировании запросов и отчетов, а также снижения риска лавинообразной
- 35 деградации производительности из-за отсутствия необходимых индексов при работе с большими объемами данных (десятки и сотни миллионов записей).

- В соответствии с первым общим аспектом данной группы изобретений электронная база данных содержит элементы базы данных, организованные в виде единственной
- 40 таблицы, при этом единственная таблица содержит по крайней мере четыре столбца, которые включают в себя: первый столбец для хранения идентификационных номеров элементов базы данных, второй столбец для хранения номеров родительских элементов для элементов базы данных, находящихся в зависимости от других элементов базы данных, третий столбец для хранения значений элементов базы данных, четвертый столбец для хранения кодов используемых типов данных; и по крайней мере пять строк,
- 45 которые включают в себя: первую строку, отображающую корневой элемент, вторую строку, отображающую тип данных, третью строку, отображающую термин, четвертую строку, отображающую реквизит термина, пятую строку, отображающую данные.

В некоторых реализациях единственная таблица дополнительно содержит пятый

столбец для хранения порядковых номеров элементов среди равных по подчиненности.

В соответствии со вторым общим аспектом данной группы изобретений способ формирования электронной базы данных исполняют с помощью системы управления базой данных посредством электронного устройства по следующим этапам: данные организуют с помощью системы управления базой данных в единственную таблицу, содержащую по крайней мере четыре столбца и по крайней мере пять строк; формируют первую строку, отображающую корневой элемент, в первом столбце которой сохраняют первый идентификационный номер; формируют вторую строку, отображающую тип данных, в первом столбце которой сохраняют уникальный идентификационный номер типа данных, при этом уникальный идентификационный номер типа данных отличается от первого идентификационного номера, в третьем столбце сохраняют наименование используемого типа данных и в четвертом столбце сохраняют код типа данных; формируют третью строку, отображающую термин, в первом столбце которой сохраняют уникальный идентификационный номер термина, в третьем столбце сохраняют наименование термина и в четвертом столбце сохраняют код используемого типа данных; формируют четвертую строку, отображающую реквизит термина, в первом столбце которой сохраняют уникальный идентификационный номер реквизита термина, а во втором столбце сохраняют идентификационный номер родительского элемента реквизита термина, при этом реквизит термина зависит от родительского элемента; и формируют пятую строку, отображающую данные, в первом столбце которой сохраняют уникальный идентификационный номер данных, во втором столбце сохраняют идентификационный номер родительского элемента данных, при этом данные зависят от родительского элемента, в третьем столбце сохраняют данные и в четвертом столбце сохраняют уникальный код типа данных.

В некоторых реализациях в способе дополнительно формируют пятый столбец для хранения порядковых номеров элементов среди элементов, равных по подчиненности.

Благодаря заявляемой группе изобретений хранение информации осуществляется в единственной индексированной таблице с определенным количеством столбцов (колонок). Таким образом, не требуется изменения самой структуры базы данных: добавление или удаление новых таблиц и связей, добавление индексов. Разработанная единожды заявляемая система становится универсальной для любых пользовательских задач и исключает стороннее вмешательство, сводя на нет ошибки, которые могут возникнуть при добавлении, изменении и удалении данных, что обеспечивает высокую надежность и стабильность системы.

Универсальность (многоплановость) заявляемой системы электронной базы данных с использованием Web-интерфейса позволяет скрыть от пользователя все технические детали базы данных и внутренних алгоритмов работы с данными и предоставить ему возможность работать с моделью данных на понятном ему языке: все объекты предметной области и действия над ними описываются без использования прикладного программирования, что также способствует наряду с повышением удобства использования повышению надежности системы.

В заявляемой электронной базе данных все данные хранятся в единственной таблице, состоящей из пяти столбцов. Эти пять столбцов - достаточный набор для построения любой структуры данных, определяющий все признаки единицы информации:

идентификатор (физическое расположение), тип, родитель, значение и порядок (среди элементов этого типа у этого родителя). Первый столбец содержит идентификационные номера элементов базы данных (ID), второй - идентификационные номера родительских элементов для элементов, находящихся в подчиненной зависимости от других элементов

(Parent ID), третий - значения элементов (Value), четвертый - код описанной в этой же таблице записи, используемый в качестве типа данного элемента (Type), пятый - порядковые номера элементов среди равных по подчиненности (Order).

Таблица 1.

Назначение	Тип	Описание
ID	Integer	Уникальный идентификационный номер
Parent ID	Integer	ID родительской записи
Type	Integer	ID типа элемента
Value	String	Значение элемента
Order	Integer	Порядок среди равных по подчиненности

Пятый столбец (Order) является дополнительным и служит для упорядочивания выборки по запросу пользователя.

Индексы, построенные для таблицы:

PRIMARY (ID) - уникальный индекс (первичный ключ);

Type + Value - тип и значение для индексированного поиска по значению;

Parent ID + Type - идентификационный номер родительского элемента и код типа элемента для поиска реквизитов элемента с использованием индекса.

Для высоконагруженных систем полезно построить еще один индекс: Type + ID - код типа элемента и идентификационный номер для ускоренного доступа к определенному типу данных.

Указанные выше индексы используются для ускорения выборки данных для нужд системы: сбора метаданных, т.е. данных, описывающих структуру данных; выборки экземпляров типов с их реквизитами и внесения изменений в данные, согласно метаданным и командам пользователя.

Тип элемента (Type) представляет собой уникальный код, идентифицирующий аналог комбинации Таблица - Поле. Наличие индекса по этому полю позволяет вести поиск в ограниченном пространстве: нет необходимости просматривать всю таблицу, а достаточно просмотреть только ту ее часть, что относится к нужному полю таблицы. На фиг. 1 изображено, как наличие индекса (Type index) позволяет компактно хранить и быстро найти список физических адресов данных в таблице данных (Data table) без необходимости просматривать всю таблицу данных.

Определение данных, взаимосвязи и подчиненности хранятся в виде метаданных в той же таблице, где и сами данные.

Каждый элемент (единица данных) занимает как минимум одну строку таблицы, в которую записаны его идентификационный номер, идентификационный номер родительского объекта, тип данных и значение (данные). Значение хранится в виде набора символов порциями ограниченной длины, например, 127 символов. В случае, когда размер поля в третьем столбце Value недостаточен для хранения значения элемента, оставшаяся часть этого значения записывается в подчиненные элементы - строки, пронумерованные по порядку заполнения в поле пятого столбца Order. Такие строки имеют специальный тип Type, равный 0, который не описан в метаданных, но активно используется в заявляемой базе данных. В случае отсутствия пятого столбца Order очередность следования фрагментов значений элементов в третьем столбце Value определяется очередностью их идентификационных номеров (ID).

Метаданные подчинены несуществующему элементу с ID=0, независимые элементы

(объекты) подчинены корневому элементу с ID=1.

Метаданные описывают структуру данных, и в соответствии с этой структурой СУБД создает, хранит и обрабатывает элементы данных.

Заявляемая база данных понимает базовые типы данных, перечисленные ниже, а также позволяет создавать новые базовые и произвольные типы данных, наследующие свойства основных типов. Базовый тип в качестве типа использует свой собственный ID. Произвольный тип в качестве типа использует ссылку на базовый тип или на другой произвольный тип.

Пример набора базовых типов данных показан в Таблице 2.

Таблица 2.

Тип	Описание
SHORT	Короткая строка (до 127 символов)
CHARS	Строка неограниченной длины
MEMO	Текст (отличается от CHARS только многострочным полем ввода)
DATE	Дата
DATETIME	Дата и время
FILE	Файл
NUMBER	Целое число
SIGNED	Число с десятичной частью
BUTTON	Действие (кнопка) над экземпляром типа
BOOLEAN	Логическое значение (Да / Нет)

Базовые типы данных используются как метаданные и идентифицируются по полю в третьем столбце Value. Запись метаданных для определения типа содержит в четвертом столбце Type идентификатор базового типа или свой собственный в случае, когда эта запись сама описывает базовый тип.

Свойства базовых типов данных, в силу ограниченного и фиксированного их набора, могут быть единожды запрограммированы и использоваться для обработки и представления данных пользователю. Набор базовых типов обеспечивает возможность создавать произвольные структуры данных любой сложности.

В качестве типа данных также можно использовать ссылку на идентификационный номер любого существующего небазового типа, что будет означать соответствующую связь двух этих типов. Таким образом можно описать любое количество взаимосвязанных типов элементов, каждый из которых будет иметь любое количество реквизитов.

Пример описания метаданных и данных в базе данных показан в Таблице 3.

Таблица 3.

Id	Parent ID	Value (комментарий)	Type	Order	Комментарий
1	1	Root	1	0	Корневой элемент
8	0	CHARS	8	0	Базовые типы данных
9	0	DATE	9	0	
11	0	BOOLEAN	11	0	
13	0	NUMBER	13	0	
15	0	SIGNED	15	0	
2	0	Договор	13	0	Термины – произвольные типы данных
3	0	Заказчик	8	0	
14	0	(ссылка на Заказчика: поле Value пустое, а в качестве типа используется ID термина, на который ссылается этот тип)	3	0	
4	0	Счет	13	0	
5	0	Сумма	15	0	
18	0	Дата	9	0	
21	0	Продукт	8	0	
22	0	Предмет договора	8	0	
23	0	Цена	15	0	
24	0	Себестоимость	15	0	
19	2	(Поле данных пустое.	18	1	Реквизиты терминов

Id	Parent ID	Value (комментарий)	Type	Order	Комментарий
		Эта запись фиксирует факт, что у объекта «Договор» (2) есть реквизит типа «Дата» (18))			
20	2	(Поле данных пустое)	22	2	
25	2	(Поле данных пустое)	23	3	
26	2	(Поле данных пустое, эта запись задает реквизит «ссылка на Заказчика (14)» у типа «Договор» (2))	14	4	
28	21	(Поле данных пустое)	23	5	
29	21	(Поле данных пустое)	24	6	
1181	1	1161 (договор)	2	1	Данные
3561	1181	20060602 (дата)	18	1	
3562	1181	Проведение работ...	22	2	
3563	1181	40000 (сумма)	23	3	
3565	1181	(ссылка на Заказчика)	113326	4	

Способ реализуют следующим образом.

С помощью СУБД, например, MySQL, посредством языка программирования SQL формируют описанную выше таблицу, начиная с записи корневого элемента в первой строке.

Затем во второй и последующей строках указывают базовые типы данных, присваивая в первом столбце ID уникальный идентификационный номер. Значениям ячеек второго столбца Parent ID присваивают нулевое значение, т.к. элементы типов данных не имеют родительских элементов. В ячейках третьего столбца Value указывают наименования типов данных. В ячейках четвертого столбца Type базовым типам данных присваивают их же ID - базовые типы ссылаются сами на себя в качестве типа. Значениям ячеек пятого столбца Order присваивают также нулевое значение в связи с независимостью данных элементов и отсутствием порядка подчиненности.

В последующих строках описывают термины, используемые пользователем для описания данных. Термин может определять ссылку на другой зарегистрированный ранее термин. В ячейках первого столбца ID указывают уникальные идентификационные номера терминов. В ячейках второго столбца Parent ID указывают нулевое значение. В ячейках третьего столбца Value указывают названия терминов или оставляют его пустым в случае, когда этот термин является ссылкой на другой термин. В ячейках четвертого столбца Type терминам присваивают код типа данного термина - ID базового типа или ID другого описанного ранее термина в случае, когда этот термин является

ссылкой на другой термин. Ячейкам пятого столбца Order присваивают нулевое значение.

В соответствии терминам описывают зависимые от них реквизиты - параметры или свойства терминов, используемые пользователем для описания данных. В ячейках первого столбца ID указывают уникальные идентификационные номера реквизитов. В ячейках второго столбца Parent ID указывают идентификационные номера родительских терминов. В ячейках третьего столбца Value может храниться дополнительная служебная информация, например, значение по умолчанию для этого реквизита, признак обязательности его наличия или маска для проверки его валидности. В ячейках четвертого столбца Type указывают тип реквизита - ID описанного ранее элемента для идентификации назначения и свойств данного термина. Ячейкам пятого столбца Order присваивают численное значение порядка подчиненности среди других равнозначных реквизитов термина.

Затем описывают непосредственно данные, содержащие смысловую информацию для пользователя и которые будут служить наполнением для отчетов при обработке запросов пользователя. В ячейках первого столбца ID указывают уникальные идентификационные номера единиц данных. В ячейках второго столбца Parent ID указывают идентификационный номер корневого элемента либо родительских терминов при их наличии. В ячейках третьего столбца Value указывают значения данных. В ячейках четвертого столбца Type указывают ID типа данных, определяющего свойства этой конкретной записи. Ячейкам пятого столбца Order присваивают численные значения порядка подчиненности данных при наличии нескольких равнозначных элементов, подчиненных другому одному элементу данных.

Человеку, имеющему навыки в данной области техники, понятно, что указанный порядок заполнения строк и столбцов дан в качестве примера и служит для целей описания сущности изобретения. В каждом конкретном случае порядок строк и столбцов может быть иным в зависимости от последовательности добавления, изменения и удаления информации из таблицы.

В контексте данного описания, SQL (structured query language) - формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД).

В контексте данного описания MySQL свободная реляционная система управления базами данных (СУБД).

В контексте данного описания веб-интерфейс - это совокупность средств, при помощи которых пользователь взаимодействует с веб-сайтом или любым другим приложением через браузер.

В контексте данного описания база данных - представленная в объективной форме совокупность самостоятельных данных.

В контексте данного описания система управления базой данных (СУБД) - совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

В контексте данного описания элемент (единица) базы данных - структурная единица базы данных, являющаяся корневым элементом, типом данных, термином, реквизитом или данными.

В контексте данного описания индекс - объект базы данных, создаваемый с целью повышения производительности поиска данных. Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и их поиск по заданному критерию путем последовательного просмотра таблицы строка за строкой

может занимать много времени. Индекс формируется из значений одного или нескольких столбцов таблицы, отсортированных по значению, и указателей на соответствующие строки таблицы и, таким образом, позволяет быстрее искать строки, удовлетворяющие критерию поиска.

5 В контексте данного описания, «сервер» представляет собой компьютерную программу, которая работает на соответствующем оборудовании и способна принимать запросы (например, от электронных устройств) по сети и выполнять эти запросы или
10 вызывать эти запросы для исполнения. Оборудование может быть представлять собой один физический компьютер или одну физическую компьютерную систему, однако это не обязательно в отношении заявляемой группы изобретений. В данном контексте
использование выражения «сервер» не означает, что каждая задача (например, полученные инструкции или запросы) или какая-либо конкретная задача будет получена, выполнена или повлекла за собой выполнение таким сервером (т.е. таким программным
15 обеспечением и/или оборудованием). Это означает, что любое количество программных или аппаратных устройств может участвовать в приеме/отправке, выполнении или послужить причиной выполнения любой задачи или запроса или последствий любой задачи или запроса. И все это программное и аппаратное обеспечение может быть одним сервером или несколькими серверами, которые включены в выражение «по крайней мере один сервер».

20 В контексте данного описания «электронное устройство» представляет собой любое компьютерное оборудование, которое может запускать программное обеспечение для соответствующей задачи. В контексте данного описания термин «электронное устройство» означает, что устройство может функционировать в качестве сервера для других электронных устройств, однако это не обязательно в отношении данной группы
25 изобретений. Таким образом, некоторыми (не ограничивающими) примерами электронных устройств являются персональные компьютеры (настольные компьютеры, ноутбуки, нетбуки и т.д.), смартфоны и планшеты, а также сетевое оборудование, такое как маршрутизаторы, коммутаторы и шлюзы. Следует понимать, что в данном контексте тот факт, что устройство функционирует как электронное устройство, не означает, что
30 оно не может функционировать в качестве сервера для других электронных устройств. Использование выражения «электронное устройство» не исключает использование нескольких электронных устройств при приеме/отправке, выполнении или вызове выполнения какой-либо задачи или запроса или последствий любой задачи или запроса или этапов любого описанного здесь способа.

35 В контексте данного описания выражение «информация» включает в себя информацию любого характера или вида, которая может быть сохранена в базе данных. Таким образом, информация включает в себя, но не ограничивается указанным перечнем: аудиовизуальные произведения (изображения, фильмы, звуковые записи, презентации и т.д.), данные (данные о местоположении, цифровые данные и т.д.), текст
40 (мнения, комментарии, вопросы, сообщения и т.д.), документы, таблицы и т.д.

В контексте данного описания выражение «программный компонент» означает наличие программного обеспечения (соответствующего конкретному аппаратному оборудованию), которое необходимо и достаточно для достижения конкретной функции (функций), на которую делается ссылка.

45 В контексте данного описания слова «первый», «второй», «третий» и т.д. использованы в качестве прилагательных только с целью указания различия между существительными, с которыми они связаны, а не для описания какой-либо конкретной связи между этими существительными. Так, например, следует понимать, что

использование терминов «первая база данных» и «третий сервер» не подразумевает какого-либо конкретного порядка, типа, хронологии, иерархии или ранжирования, например между серверами. Их использование (само по себе) не подразумевает, что

любой «второй сервер» обязательно должен существовать в любой конкретной ситуации. Кроме того, как обсуждается здесь в других контекстах, ссылка на «первый» элемент и «второй» элемент не исключает того, что два элемента являются одним и тем же фактическим элементом реального мира. Таким образом, например, в некоторых случаях «первый» сервер и «второй» сервер могут быть одними и теми же программными и/или аппаратными компонентами, в других случаях они могут быть различными программными и/или аппаратными компонентами.

Реализации заявляемой группы изобретений имеют по меньшей мере один из вышеупомянутых объектов и/или аспектов, но необязательно содержат все из них. Следует понимать, что некоторые аспекты настоящей группы изобретений, которые возникли в результате попыток достичь вышеупомянутого объекта, могут не удовлетворять этому объекту и/или могут удовлетворять другим объектам, не указанным здесь конкретно.

Дополнительные и/или другие особенности, аспекты и преимущества реализаций данной группы изобретений станут очевидными из нижеследующего описания, прилагаемых чертежей и прилагаемой формулы изобретения.

КРАТКОЕ ОПИСАНИЕ ИЛЛЮСТРАЦИЙ

Эти и другие особенности, аспекты и преимущества заявляемой группы изобретений станут более понятными на основе нижеследующего описания, прилагаемой формулы изобретения и сопроводительных чертежей, где:

Фиг. 1 иллюстрирует табличное представление типа индекса и данных в таблице.
 Фиг. 2 иллюстрирует табличное представление массива данных «Клиент» и «Встреча».
 Фиг. 3 иллюстрирует табличное представление массива данных «Договор».
 Фиг. 4 иллюстрирует вид редактора по вкладке «Типы».
 Фиг. 5 иллюстрирует соответствие полей редактора полям табличного представления.
 Фиг. 6 иллюстрирует форму для формирования отчета.
 Фиг. 7 иллюстрирует вид сформированного отчета.

ДЕТАЛЬНОЕ ОПИСАНИЕ ИЛЛЮСТРАЦИЙ

Приведенные здесь примеры и условный язык в основном предназначены для того, чтобы помочь читателю понять принципы заявляемой группы изобретений и не ограничивать ее сферу применения указанными примерами и условиями. Понятно, что специалисты в данной области техники могут разработать различные устройства, которые, хотя они явно не описаны или показаны здесь, тем не менее, воплощают принципы заявляемой группы изобретений и включены в ее сущность и сферу применения.

Кроме того, для облегчения понимания следующее описание может описывать относительно упрощенные реализации заявляемой группы изобретений. Как понятно специалистам в данной области техники, различные реализации заявляемой группы изобретений могут иметь большую сложность.

В некоторых случаях также могут быть изложены полезные примеры модификаций заявляемой группы изобретений. Это делается для облегчения понимания и, опять же, не для определения сферы применения или определения границ заявляемой группы изобретений. Эти модификации не являются исчерпывающим списком, и специалист в данной области техники может внести другие изменения, оставаясь при этом в рамках заявляемой группы изобретений. Кроме того, если не было представлено примеров

модификаций, не следует понимать, что никакие модификации невозможны и/или то, что описано, является единственным способом реализации заявляемой группы.

Кроме того, все заявления, приводимые здесь, в которых перечислены принципы, аспекты и реализации заявляемой группы изобретений, а также их конкретные примеры, предназначены для охвата как их структурных, так и функциональных эквивалентов, независимо от того, известны ли они в настоящее время или разработаны в будущем. Таким образом, например, специалистам в данной области техники будет понятно, что любые блок-схемы здесь представляют концептуальные представления иллюстративных схем, воплощающих принципы заявляемой группы изобретений. Также понятно, что любые блок-схемы, диаграммы, псевдокод и т.п. представляют различные процессы, которые могут быть в основном представлены на машиночитаемых носителях и, таким образом, выполняться компьютером или процессором, независимо от того, являются ли они явно отображаемыми компьютером или процессором.

На фиг. 1-7 использованы следующие обозначения: 1 - термины, 2 реквизиты и 3 - данные.

СУБД и сама база данных размещены на сервере и доступны пользователю посредством сети Интернет через Web-интерфейс, решение также может использоваться локально на компьютере пользователя.

После регистрации в заявляемой системе посредством сети Интернет с использованием браузера, установленного на стандартном персональном компьютере, пользователь получает авторизованный профиль с доступом к управлению базой данных. Пользователь, его профиль и ограничения его доступа к данным описаны в этой же базе данных средствами заявляемой системы.

На фиг. 2 и 3 показаны табличные представления массива данных, заполняемых пользователем, на которых «Клиент», «Встречи» и «Договор» являются терминами 1, наименования «Клиент», «Категория», «Телефон», «Дата» и т.д. - реквизитами 2 терминов 1, а в ячейках таблицы указаны сами данные 3.

Основная проблема информационных систем - это растущая цена их поддержки при возрастании сложности системы, причем зачастую затраты на разработку и сопровождение растут экспоненциально (см. Фиг. 8). Два основных фактора риска здесь: несовершенство разработки программного кода и компромисс при выборе архитектуры базы данных. Заявляемая система позволяет пользователю самостоятельно задать сколько угодно сложную структуру данных и правила их обработки без необходимости создания таблиц базы данных, индексов, хранимых процедур и функций, программного кода. Таким образом, нет необходимости обращаться к дорогостоящей команде разработки: аналитик, ведущий разработчик, программист/кодер, тестировщик, внедренец. Заявляемая группа изобретений используют уже готовый программный код и архитектурное решение в виде ограниченного набора примитивных блоков, из которых можно построить информационную систему любой сложности, обеспечивая универсальность системы, в которой зависимость затрат на сопровождение от сложности системы будет не хуже линейной. Это преимущество обеспечивается определенной избыточностью данных, поскольку в ней проиндексированы все хранящиеся данные, и вычислений, однако по стоимости эти затраты намного ниже временных и материальных затрат на содержание команды разработки. Самый главный риск - человеческий фактор при проектировании и реализации программного кода и базы данных - полностью исключен.

Пользователь имеет все средства для добавления, выборки, изменения и удаления данных, которые доступны в любой платформе прикладного программирования.

Система позволяет создавать произвольные шаблоны и формы интерфейса пользователя, наполнять их данными, выбранными по произвольным правилам, изменять данные и сохранять их в базе данных и таким образом формировать интерфейс и логику прикладной программы, не прибегая собственно к программированию.

Иными словами, заявляемая группа изобретений предоставляют законченное средство разработки прикладного программного обеспечения.

На фиг. 4 показаны поля редактора справочника типов, позволяющего пользователю задавать типы данных и их реквизиты (свойства), создавая специфическую иерархию данных для каждой конкретной ситуации на основе единственной таблицы заявляемой системы. Пользователь видит все типы, которым обучена система, а также их подчиненности и связи. В частности, он видит, что тип Клиент имеет реквизиты Телефон, Категория и Встречи. При этом Категория - это справочное значение, которое определяется ссылкой на запись в справочнике Категория; Встречи - это подчиненный массив записей. Заявляемая система позволяет представить все это пользователю без раскрытия технических деталей, он оперирует только терминами своего бизнеса и двумя простейшими мнемониками: префикс ссылки на справочник «-->» и базовый тип данных (DATE, CHARS, NUMBER и другие). Достаточный уровень подготовки для работы со справочником типов - это грамотный пользователь Microsoft Excel, что в настоящее время доступно большинству офисных работников.

На фиг. 5 показано соответствие полей редактора типов полям табличного представления. Редактор типов позволяет создать понятное пользователю представление бизнес-сущностей в виде таблиц. Пользователь может настроить порядок колонок в таблице (значок «^» в редакторе типов перемещает реквизит на одну строку выше) и задать обязательные к заполнению реквизиты типа (восклицательный знак в квадратных скобках «[!]» означает, что реквизит обязателен к заполнению). Структура, заданная в редакторе типов, отображается соответствующим образом в таблицах справочника заявляемой системы.

На фиг. 6 и 7 показаны форма для формирования отчета и его вид, соответственно. Построение отчетов заключается в выборе произвольного количества колонок отчета из всего доступного множества типов. Система позволяет выбрать только связанные типы, при этом от пользователя не требуется указывать, как именно связать выбранные им типы - система сделает это самостоятельно, используя связи, заданные в редакторе типов в виде ссылок и подчиненных массивов. Функционал построителя отчетов позволяет реализовать основные функции выборки, фильтрации и агрегации данных, которые существуют в языке SQL, включая вложенные отчеты. Также при построении отчетов можно использовать все встроенные функции реляционной базы данных, на которой реализована система.

Назначение полей таблицы Колонки отчета следующее.

Имя в отчете - название колонки в выводимом отчете, по умолчанию - значение Колонки отчета.

Формула - задает вычисляемое выражение или псевдоним колонки. В выпадающем списке колонок отчета на последней позиции находится значение «Вычисляемое». Это синтетическое поле, которое позволяет вставить произвольные операторы и функции, допустимые в языке SQL. При этом можно оперировать любыми другими колонками отчета, если для них задан псевдоним в их поле Формула.

Если в поле Формула встречается выражение в квадратных скобках, то система попытается найти соответствующий отчет и использовать данные из него, как из вложенного отчета.

Значение (от), Значение (до) - диапазон значений (для дат и чисел) или маска (для текста). Также здесь допускается использовать значение других полей отчета, если заключить их имя из колонки Формула в квадратные скобки.

5 Функция - агрегирующая функция, применяемая к значению поля: AVG (среднее), COUNT (количество), MAX (максимальное), MIN (минимальное), SUM (сумма). Группировка значений, посчитанных этой функцией, происходит по всем полям, в которых не используется агрегирование (не указана одна из перечисленных функций).

10 Функция (от), Функция (до) - диапазон значений (для дат и чисел) или маска (для текста), применяемые после вычисления функций, указанных в поле Функция (аналог ключа HAVING в SQL). Здесь допускается использовать значение других полей отчета, если заключить их имя из колонки Формула в квадратные скобки.

Скрыть - если поставить галку в этом поле, то соответствующая колонка не будет отображена в отчете, хотя ее значение может использоваться для вычислений, фильтрации и сортировки отчета.

15 Формат - конечный формат отображения поля. Служит для приведения значения к нужному виду.

Сорт. - указатель последовательности и направления сортировки отчета.

Отрицательное число здесь даст обратную сортировку (по убыванию значений).

20 Итог - агрегирующая функция, применяемая к значениям колонки отчета. Если указана функция хотя бы в одной колонке, то в отчете выводится итоговая строка с соответствующими значениями.

Заявляемая группа изобретений позволяет на основе универсальной системы в доступном для пользователя виде организовывать индивидуальную для каждого случая базу данных, проявляющую высокую надежность при использовании.

25 Должно быть ясно понято, что не все технические эффекты, упомянутые в данном документе, необходимо использовать в каждом варианте заявляемой группы изобретений. Например, варианты осуществления заявляемой группы изобретений могут быть реализованы без использования пользователем некоторых из этих технических эффектов, в то время как другие варианты осуществления могут быть
30 реализованы пользователем с использованием других технических эффектов или вообще без них.

Модификации и усовершенствования вышеописанных реализаций заявляемой группы изобретений могут стать очевидными для специалистов в данной области техники. Вышеприведенное описание должно быть примерным, а не ограничивающим. Таким
35 образом, объем заявляемой группы изобретений ограничивается только объемом прилагаемой формулы изобретения.

(57) Формула изобретения

1. Способ формирования электронной базы данных, характеризующийся тем, что
40 способ исполняют с помощью системы управления базой данных посредством электронного устройства по следующим этапам:

данные организуют с помощью системы управления базой данных в единственную таблицу, содержащую по крайней мере четыре столбца и по крайней мере пять строк;

45 формируют первую строку, отображающую корневой элемент, в первом столбце которой сохраняют первый идентификационный номер;

формируют вторую строку, отображающую тип данных, в первом столбце которой сохраняют уникальный идентификационный номер типа данных, при этом уникальный идентификационный номер типа данных отличается от первого идентификационного

номера, в третьем столбце сохраняют наименование используемого типа данных и в четвертом столбце сохраняют код типа данных;

формируют третью строку, отображающую термин, в первом столбце которой сохраняют уникальный идентификационный номер термина, в третьем столбце сохраняют наименование термина и в четвертом столбце сохраняют код используемого типа данных;

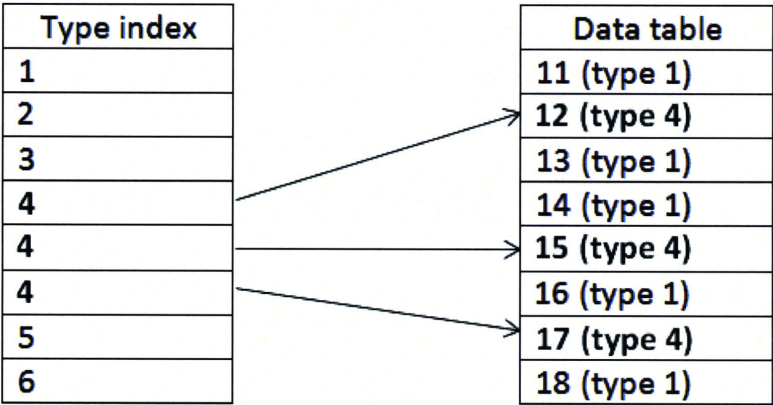
формируют четвертую строку, отображающую реквизит термина, в первом столбце которой сохраняют уникальный идентификационный номер реквизита термина, а во втором столбце сохраняют идентификационный номер родительского элемента реквизита термина, при этом реквизит термина зависит от родительского элемента; и

формируют пятую строку, отображающую данные, в первом столбце которой сохраняют уникальный идентификационный номер данных, во втором столбце сохраняют идентификационный номер родительского элемента данных, при этом данные зависят от родительского элемента, в третьем столбце сохраняют данные и четвертом столбце сохраняют уникальный код типа данных.

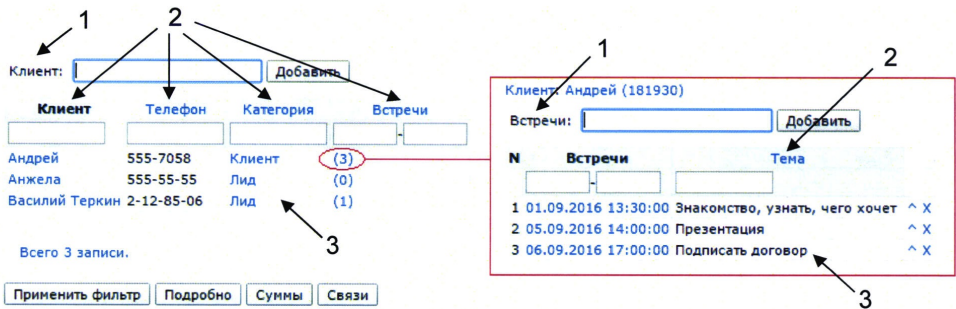
2. Способ по п. 1, характеризующийся тем, что дополнительно формируют пятый столбец для сохранения порядковых номеров элементов среди элементов, равных по подчиненности.

1

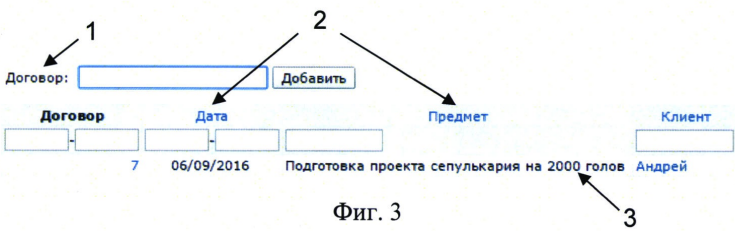
ЭЛЕКТРОННАЯ БАЗА ДАННЫХ И СПОСОБ ЕЁ ФОРМИРОВАНИЯ



Фиг. 1



Фиг. 2



Фиг. 3

2

ЭЛЕКТРОННАЯ БАЗА ДАННЫХ И СПОСОБ ЕЁ ФОРМИРОВАНИЯ

[Словарь](#)
[Типы](#)
[Файлы](#)
[Выход](#)

Встречи DATETIME

Реквизиты: Значение по умолчанию:

Тема x ^ []

Дата DATE

Реквизиты: Значение по умолчанию:

Договор NUMBER

Реквизиты: Значение по умолчанию:

Дата x ^ []

Предмет x ^ []

--> Клиент x ^ []

--> Категория

Реквизиты: Значение по умолчанию:

Категория SHORT

Реквизиты: Значение по умолчанию:

--> Клиент

Реквизиты: Значение по умолчанию:

Клиент SHORT

Реквизиты: Значение по умолчанию:

Телефон x ^ []

--> Категория x ^ []

Встречи x ^ []

Телефон CHARS

Реквизиты: Значение по умолчанию:

Тема SHORT

Реквизиты: Значение по умолчанию:

Предмет MEMO

Реквизиты: Значение по умолчанию:

Новый Тип:

Новая ссылка на Тип:

Фиг. 4

ЭЛЕКТРОННАЯ БАЗА ДАННЫХ И СПОСОБ ЕЁ ФОРМИРОВАНИЯ

Клиент	Клиент	Телефон	Категория	Встречи
Реквизиты:				
Телефон	Андрей	555-7058	Клиент	(3)
--> Категория	Анжела	555-55-55	Лид	(0)
Встречи	Василий Теркин	2-12-85-06	Лид	(1)
	Эраст Неизвестный			(0)

Фиг. 5

Отчет: Встречи за тек. месяц (181964)

Колонки отчета: Встречи

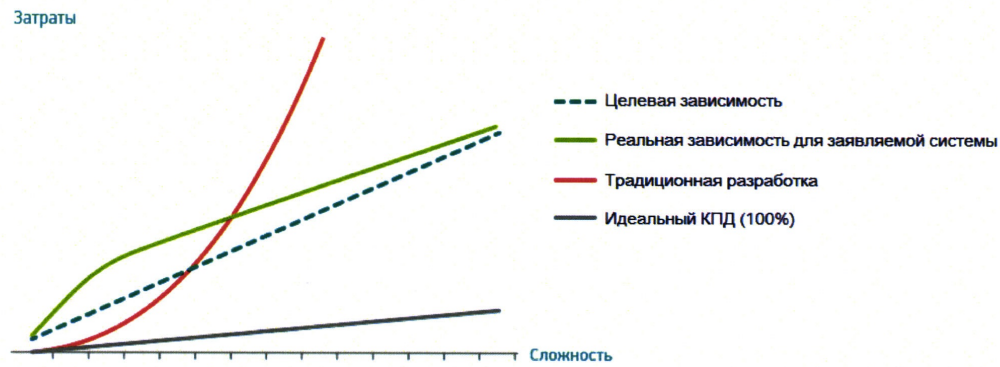
N	Колонки отчета	Имя в отчете	Формула	Значение (от)	Значение (до)	Функция	Функция (от)	Функция (до)	Скрыть	Формат	Сорт.	Итог
1	Клиент											COUNT ^ X
2	Клиент -> Категория	Категория										^ X
3	Договор		%						X			^ X
4	Клиент -> Телефон	Телефон										^ X
5	Встречи	Дата	1		[TODAY]							^ X
6	Встречи -> Тема	Тема встречи										^ X

Фиг. 6

Встречи за тек. месяц

Клиент	Категория	Телефон	Дата	Тема встречи
Андрей	Клиент	555-7058	01.09.2016 13:30:00	Знакомство, узнать, чего хочет
Андрей	Клиент	555-7058	05.09.2016 14:00:00	Презентация
Андрей	Клиент	555-7058	06.09.2016 17:00:00	Подписать договор
3				

Фиг. 7



Фиг. 8